

**PENGEMBANGAN APLIKASI E-COMMERCE
DENGAN KONSEP BERORIENTASI OBJEK
PADA PT SHARP CABANG SEMARANG**

Arifian Rahman¹, Suhartono², dan Nurdin Bahtiar

Program Studi Teknik Informatika Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Diponegoro

fian_nerazzuri@yahoo.co.id

Abstract

PT Sharp Semarang branch is engage in the marketing of electronic items like air conditioners, refrigerators, televisions, and others. Business competition is increasingly fierce, especially in the electronics field requires the selection of advertising media that can market their goods quickly and easily. E-commerce as a new method that uses the Internet media is the right solution to market their goods easily and quickly. Application E-commerce Sharp Semarang developed using method of object-oriented software engineering (OOSE / Object Oriented Software Engineering). By using the method of object-oriented software engineering Application E-commerce Sharp Semarang is expected to has more quality in terms of functionality and the stage of develope.

Keywords : E-commerce, Application E-commerce Sharp Semarang, OOSE, object

1. Pendahuluan

Dengan berkembangnya zaman membuat persaingan di dunia bisnis semakin ketat. Untuk memperkuat bisnisnya banyak perusahaan mempergunakan media-media periklanan untuk memasarkan barang dagangannya. Media yang umumnya sering digunakan dalam periklanan antara lain adalah media cetak dan media elektronik. Perkembangan teknologi informasi yang sangat pesat membuat beralihnya media periklanan ke media Internet. Internet merupakan suatu cara baru yang lebih menyenangkan dengan menggunakan *web* sebagai basis untuk aplikasi bisnis yang lebih luas, sehingga memperluas cara perusahaan dalam berkomunikasi dan meningkatkan kemampuan perusahaan untuk saling berbagi informasi.

Media periklanan merupakan hal yang penting bagi perusahaan yang bergerak dalam bisnis perdagangan barang. Permasalahannya terletak pada penentuan media periklanan yang digunakan. Dengan jumlah barang yang cukup banyak akan lebih baik memilih Internet sebagai media periklanan. Apalagi persaingan perdagangan barang elektronik berkembang sangat pesat dalam kehidupan masyarakat. Tujuan menggunakan media Internet agar dapat membantu dalam memberikan pengenalan barang secara tepat yang tidak memandang ruang dan waktu. Strategi pemasaran ini banyak dipakai karena memiliki peran penting dalam dunia bisnis dan memiliki jangkauan pasar yang luas.

PT Sharp cabang Semarang bergerak dalam bidang pemasaran barang-barang elektronik seperti AC, lemari es, televisi, dan lain-lain. PT Sharp didirikan pada tahun 1969 dengan nama PT Yasonta. Pada tanggal 1 Mei 2005 berganti nama menjadi PT Sharp Electronics Indonesia yang dikenal sampai sekarang. PT Sharp berdedikasi pada dua prinsip yang ideal yaitu, kesungguhan hati dan kreatifitas. Dengan berpegang teguh pada dua prinsip tersebut, pelanggan dapat memperoleh kepuasan sejati, sekaligus memberikan kontribusi yang berarti bagi masyarakat (Profil Perusahaan PT Sharp. Diakses dari www.sharp-indonesia.com/new/tentang-sharp/profil-perusahaan.html). PT Sharp cabang Semarang dalam memasarkan produknya menggunakan media televisi, media radio, media cetak, spanduk, dan SPG (*Sales Promotion Girl*), sehingga membutuhkan biaya yang besar untuk memasarkan produknya.

Electronic commerce (e-commerce) merupakan salah satu metode baru dalam berbisnis melalui Internet. *E-commerce* dapat menghubungkan penjual dan pembeli yang berbeda tempat dan tidak akan menjadi kendala dalam melakukan transaksi. Calon pembeli dapat mengetahui info mengenai produk seperti harga, model, warna, dan sebagainya. Masalah tersebut yang menarik diangkat menjadi topik dalam tugas akhir ini karena dengan dibangunnya *e-commerce* ini maka akan semakin meningkatkan kinerja perusahaan.

Tugas akhir ini akan merancang dan membangun *e-commerce* yang dapat menampilkan produk-produk yang dijual oleh PT Sharp cabang Semarang serta dapat memberikan fungsi transaksi penjualan barang dengan cepat dan mudah. Aplikasi E-commerce Sharp Semarang akan dikembangkan dengan metode rekayasa perangkat lunak berorientasi objek (OOSE/*Object Oriented Software Engineering*). Dengan menggunakan metode rekayasa perangkat lunak berorientasi objek diharapkan Aplikasi E-commerce Sharp Semarang memiliki kualitas lebih dilihat dari sisi fungsionalitas dan tahap pembuatannya.

2. Tinjauan Pustaka

Tinjauan Pustaka menguraikan beberapa pengertian dasar tentang *e-commerce*, Internet, WWW, bahasa pemrograman, konsep berorientasi objek, model proses berorientasi objek, UML, dan sistem manajemen basis data MySQL.

2.1. E-commerce

E-commerce memungkinkan transaksi dilakukan menggunakan teknologi digital. Secara garis besar *e-commerce* adalah transaksi komersial yang terjadi melalui Internet dan *web* yang meliputi pertukaran nilai (misalnya: uang) dengan barang atau jasa antar organisasi atau perorangan. Pertukaran nilai merupakan hal penting dalam memahami *e-commerce*. Tanpa adanya pertukaran nilai, tidak ada perdagangan yang terjadi [7].

2.2. Internet

Interconnection networking atau yang lebih populer dengan sebutan Internet, adalah sebuah sistem komunikasi global yang menghubungkan komputer-komputer dan jaringan-jaringan komputer di seluruh dunia [14]. Internet juga dapat diartikan hubungan berbagai komputer dan berbagai tipe komputer yang membentuk sistem jaringan yang mencakup seluruh dunia menggunakan jalur telekomunikasi seperti *telepon*, *wireless*, dan lainnya. Internet memungkinkan masyarakat untuk memperoleh informasi dan layanan dengan cepat. Di dalamnya terdapat berbagai macam informasi dan layanan, seperti surat elektronik, percakapan langsung, dan halaman *web* yang saling berhubungan [19].

2.3. World Wide Web

World Wide Web (WWW) atau *web* merupakan salah satu fasilitas yang disediakan di Internet. *Web* merupakan dunia maya di Internet yang terdapat ribuan info tentang segala hal dan dikemas dalam bentuk dokumen *hypertext*. *Hypertext* merupakan konsep dasar dari *web*.

Dokumen *hypertext* adalah salah satu dokumen yang memungkinkan menjelajahi dari satu halaman *web* ke halaman *web* yang lainnya dengan menggunakan suatu *hyperlinks* [11].

2.4. Bahasa Pemrograman

Mengingat Aplikasi *e-commerce* yang dibangun berbasiskan *web*, maka bahasa pemrograman yang digunakan berupa bahasa *scripting* seperti HTML, PHP, CSS, dan Javascript.

2.4.1. HTML

HTML (*Hypertext Markup Language*) merupakan sebuah bahasa *scripting* yang berguna untuk menuliskan halaman *web*. Pada halaman *web*, HTML dijadikan sebagai bahasa *scripting* dasar yang berjalan bersama berbagai bahasa *scripting* pemrograman lainnya [11].

2.4.2. PHP

PHP adalah bahasa *scripting* yang menjadi satu dengan HTML yang dijalankan pada *server side*, artinya adalah semua perintah yang diberikan akan secara penuh dijalankan pada *server* sedangkan yang dikirimkan ke *client (browser)* hanya berupa hasilnya saja. Kebanyakan sintaks pada PHP mengadopsi sintaks bahasa pemrograman C, JAVA, PERL dengan pasangan fitur khusus untuk PHP yang bersifat unik [6].

2.4.3. CSS

CSS merupakan singkatan dari *Cascading Style Sheet*. CSS adalah sebuah dokumen yang berguna untuk melakukan pengaturan pada komponen halaman *web*, inti dari dokumen ini adalah memformat halaman *web* standar menjadi bentuk *web* yang memiliki kualitas yang lebih indah dan menarik [10].

2.4.4. Javascript

Javascript merupakan bahasa *scripting* yang digunakan dalam membuat sebuah *website* yang berfungsi memberi suatu aksi pada tag HTML. *Javascript* biasanya digunakan dalam merancang antarmuka halaman agar lebih interaktif [3].

Javascript merupakan bahasa yang berbentuk kumpulan *scripting* yang pada fungsinya berjalan pada satu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan eksekusi perintah-perintah disisi pengguna, yang artinya disisi *browser* bukan disisi *server web* [3].

2.5. Konsep Berorientasi Objek

Ada banyak cara untuk menyelesaikan suatu permasalahan menggunakan solusi berbasis perangkat lunak (*software-based solution*). Salah satu yang umum digunakan untuk menyelesaikan permasalahan adalah dengan *object-oriented viewpoint*. Permasalahan utama digolongkan sebagai sebuah kumpulan objek yang mempunyai sifat (atribut) dan tingkah laku (behavior) khusus. Objek-objek tersebut dimanipulasi dengan kumpulan fungsi-fungsi (yang disebut *methods*, *operations*, atau *services*) dan objek-objek tersebut saling berinteraksi satu sama lain [12].

2.5.1. Kelas (*Class*), Objek (*Object*), dan Atribut (*Attribute*)

Kelas adalah suatu konsep berorientasi objek yang mengenkapsulasi data dan abstraksi prosedural yang dibutuhkan untuk menggambarkan isi dan tingkah laku entitas-entitas yang ada di dunia nyata [12]. Sebuah objek adalah suatu entitas yang memiliki kumpulan atribut objek tersebut dan operasi-operasi yang dijalankan pada kumpulan atribut-atribut objek. Operasi-operasi tersebut terkait dengan penyediaan *services* terhadap objek lain (klien) yang *me-request services* itu ketika dibutuhkan suatu komputasi [16]. Inggriani Liem dari ITB mendefinisikan objek adalah abstraksi dari sesuatu yang mewakili sesuatu pada dunia nyata, sedangkan kelas merupakan kumpulan objek yang memiliki atribut yang sama [9].

2.5.1.1. Identifikasi Kelas/Objek

Kelas/Objek dapat diidentifikasi dengan menganalisis permasalahan atau dengan melakukan analisis pada deskripsi naratif proses untuk sistem yang akan dibuat. Objek dapat ditentukan dari kata benda atau klausa kata benda pada deskripsi naratif tersebut [12]. Objek dapat berupa:

1. *External entities* (contoh : sistem lain, perangkat, atau orang) yang menghasilkan atau mengambil informasi yang digunakan oleh sistem berbasis komputer.
2. *Things* (contoh : laporan, tampilan, surat, sinyal) yang merupakan bagian dari informasi pada lingkup permasalahan.
3. *Occurrences* atau *events* merupakan aktivitas yang terjadi dalam konteks pengoperasian sistem.
4. *Roles* (contoh : manajer, teknisi) yang dimainkan oleh orang yang berinteraksi dengan sistem.
5. *Organizational units* (contoh : divisi, kelompok, tim) yang terkait dengan aplikasi.

6. *Places* yang menentukan konteks masalah dan fungsi keseluruhan sistem.

7. *Structures* (contoh : sensor, kendaraan roda empat) yang mendefinisikan kelas dari objek-objek.

2.5.1.2. Menetapkan Atribut

Atribut merupakan kumpulan fitur atau sifat dari suatu kelas, yaitu informasi tentang kelas yang harus dipelihara untuk mencapai tujuan perangkat lunak yang telah dispesifikasikan oleh pengguna. Atribut dapat diekstraksi dari pernyataan ruang lingkup atau dibedakan dari pemahaman tentang sifat alami sebuah kelas [12]. Atribut menggambarkan suatu objek yang telah dipilih untuk model analisis. Pada intinya, atributlah yang mendefinisikan objek, menjelaskan tujuan objek dalam konteks ruang lingkup masalah.

Untuk mengembangkan kumpulan atribut yang berarti untuk suatu objek, pengembang aplikasi dapat sekali lagi mempelajari narasi pemrosesan permasalahan dan memilih hal/benda yang layak menjadi atribut suatu objek [12].

2.5.2. *Operations, Methods, atau Services*

Suatu objek mengenkapsulasi data (representasi dari kumpulan atribut-atribut) dan algoritma yang memproses data. Algoritma inilah yang disebut *operations*, *methods*, atau *services* dan dapat dilihat sebagai modul-modul pada rekayasa perangkat lunak yang menggunakan metode konvensional. Semua kata kerja memungkinkan untuk menjadi kandidat *operations*, *methods*, atau *services* [12].

2.5.3. Enkapsulasi

Prinsip enkapsulasi sebenarnya adalah prinsip untuk menghilangkan pembatas antara data dan algoritma pemroses data itu sendiri dengan menggabungkan keduanya dalam suatu kelas atau objek. Prinsip ini lalu menghasilkan suatu pengertian yaitu *information hiding*. *Information Hiding* merupakan menyembunyikan detail data dan algoritma pemroses data kedalam suatu *interface*, sehingga yang berinteraksi dengan dunia luar hanyalah *interface*-nya saja [12].

2.5.4. *Inheritance*

Inheritance merupakan salah satu kunci pembeda antara sistem konvensional dan berorientasi objek [12]. *Inheritance* atau pewarisan adalah prinsip pewarisan sifat dari orang tua ke anak atau turunannya yang diterapkan pada kelas. Orang tua memiliki atribut dan *method* yang lebih umum dibandingkan dengan anak atau

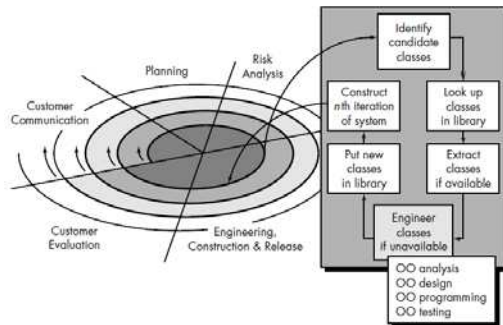
turunannya. Semua atribut dan method dari kelas leluhur akan diwariskan kepada kelas turunan yang secara intrinsik menjadi bagian dari kelas turunan atau kelas anak [12].

2.5.5. Polymorphism

Polymorphism mungkin merupakan konsep pemrograman berorientasi objek yang paling sulit dimengerti. *Polymorphism* berasal dari kata *poly* yang berarti banyak dan *morph* yang berarti bentuk. *Polymorphism* merupakan implementasi yang berbeda dari method suatu objek, namun method objek tersebut memiliki bentuk yang sama dengan objek lain [12]. Sebagai contoh ada suatu kelas hewan, kemudian didefinisikan beberapa objek yang mewarisi atribut dan *method* dari kelas hewan yaitu objek sapi, burung, dan ular. Ketiga objek tersebut memiliki *method* yang sama yaitu “Bergerak”. Namun implementasi *method* “Bergerak” ketiganya berbeda satu sama lain yaitu, sapi berjalan, burung terbang, dan ular melata.

2.6. Model Proses Berorientasi Objek

Secara umum tahap pengembangan aplikasi dengan metode rekayasa perangkat lunak berorientasi objek sebagai Model Proses Berorientasi Objek dapat dilihat pada gambar 2.1 [12].



Gambar 2.1. Model Proses OO

Model proses OO dimulai dengan *customer communication* yang mendefinisikan lingkup permasalahan. *Planning* dan *Risk Analysis* membangun pondasi untuk perencanaan pengembangan perangkat lunak. Proses selanjutnya adalah *engineering, construction & release* yaitu tahap pengembangan perangkat lunak secara teknis. Pada proses ini, serangkaian tahap mulai dari analisis, desain, implementasi, dan testing berorientasi objek dilakukan secara *iterative* dan *evolutionary*. *Iterative* maksudnya adalah setiap tahap pada proses tersebut dilakukan secara berulang, sedang *evolutionary* maksudnya adalah tahap-tahap yang dilakukan akan semakin sulit seiring iterasi (perulangan)

yang dilakukan. Inilah yang membedakan model proses OO dengan model proses spiral. Model proses OO tersebut merupakan model yang paling cocok untuk metode rekayasa perangkat lunak berorientasi objek [12].

2.7. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa standar untuk penulisan mengenai gambaran umum dari perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artifak dari sistem perangkat lunak [4]. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

2.8. Sistem Manajemen Basis Data MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*), sehingga setiap orang bebas untuk menggunakan MySQL. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [17].

3. Definisi Kebutuhan Dan Analisis Berorientasi Objek

Pembuatan perangkat lunak Aplikasi *E-commerce* pada PT Sharp cabang Semarang (AESS) dimulai dengan *customer communication* yang menghasilkan suatu batasan permasalahan. Kemudian dari lingkup permasalahan tersebut dilakukan analisis berorientasi objek. Analisis berorientasi objek diawali dengan mendefinisikan kebutuhan perangkat lunak (tahap *requirement*) kemudian memodelkan permasalahan dalam beberapa model, antara lain *use case diagram*, model CRC (*Class-Responsibility-Collaborator*) *index card* dan *class diagram analysis* yang dimodelkan dengan menggunakan *Unified Modeling Language* (UML).

3.1. Customer Communication

Suatu *perangkat* lunak yang baik atau sukses dibuat adalah perangkat lunak yang dapat memenuhi spesifikasi yang dibutuhkan oleh pengguna perangkat lunak, memiliki performa yang sempurna dalam jangka waktu yang lama, mudah dimodifikasi dan bahkan mudah untuk digunakan, serta dapat mengubah sesuatu menjadi lebih baik [12]. Sebaliknya, suatu perangkat lunak dikatakan gagal apabila kepuasan pengguna tidak tercapai, banyak terjadi

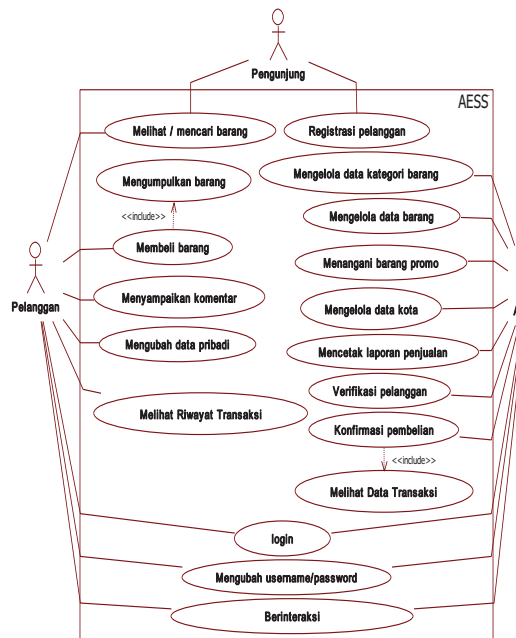
error, sulit untuk dimodifikasi atau bahkan sulit untuk digunakan, serta sesuatu yang buruk bisa terjadi karena penggunaan perangkat lunak [12].

3.2. Analisis Berorientasi Objek (OOA – Object Oriented Analysis)

OOA memodelkan permasalahan dalam beberapa model, antara lain *use case diagram*, model CRC (*Class-Responsibility-Collaborator*) *index card* dan *class diagram analysis*. Definisi kebutuhan dilakukan pada tahap *requirement* dengan menganalisis batasan permasalahan yang telah didefinisikan selama tahap *customer communication*.

3.2.1. Use Case Diagram

Model diagram *use case* AESS ditunjukkan pada gambar 3.1.



Gambar 3.1. Use case diagram AESS

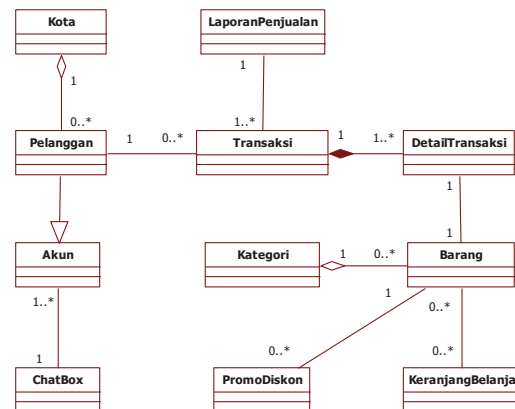
3.2.2. Class Diagram Analysis

Untuk membuat *class diagram* diawali dengan menentukan *relationships* antarkelas. Kemudian hubungan *aggregation*, *association*, dan *composition* ditentukan *multiplicity*-nya. Dalam *class diagram analysis*, *relationships* antar kelas mengacu pada model CRC *index card*. Sedangkan *multiplicity*-nya ditentukan sebagai berikut :

1. Hubungan kelas Kategori dengan kelas Barang merupakan hubungan *aggregation*, dengan *multiplicity* 1-0..*. Diasumsikan bahwa satu Kategori dapat tidak memiliki barang atau memiliki banyak barang.

2. Hubungan kelas Kota dengan kelas Pelanggan merupakan hubungan *aggregation*, dengan *multiplicity* 1-0..*. Diasumsikan bahwa satu kota dapat tidak memiliki pelanggan atau memiliki banyak pelanggan.
3. Hubungan kelas Transaksi dengan kelas DetailTransaksi merupakan hubungan *composition*, dengan *multiplicity* 1-1..*. Diasumsikan bahwa satu transaksi pembelian memiliki minimal satu detail transaksi.
4. Hubungan kelas Barang dengan kelas PromoDiskon merupakan hubungan *association*, dengan *multiplicity* 1-0..*. Diasumsikan bahwa satu barang dapat tidak memiliki promo atau memiliki banyak promo.
5. Hubungan kelas Pelanggan dengan kelas Transaksi merupakan hubungan *association*, dengan *multiplicity* 1-0..*. Diasumsikan bahwa satu pelanggan dapat tidak memiliki atau memiliki banyak transaksi pembelian.
6. Hubungan kelas ChatBox dengan kelas Akun merupakan hubungan *association*, dengan *multiplicity* 1-1..*. Diasumsikan bahwa satu chat box memiliki minimal satu akun.
7. Hubungan kelas Barang dengan kelas KeranjangBelanja merupakan hubungan *association*, dengan *multiplicity* 0..*-0..*. Diasumsikan bahwa satu barang dapat tidak memiliki keranjang belanja atau memiliki banyak keranjang belanja dan sebaliknya.
8. Hubungan kelas LaporanPenjualan dengan kelas Transaksi merupakan hubungan *association*, dengan *multiplicity* 1-1..*. Diasumsikan bahwa satu laporan penjualan memiliki minimal satu transaksi.

Setelah *multiplicity* ditentukan, hubungan antar class digambarkan dalam sebuah *class diagram analysis* seperti pada gambar 3.2.



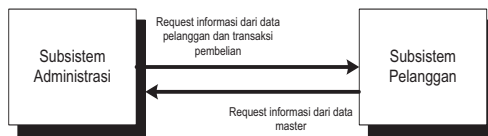
Gambar 3.2. Class Diagram Analysis

4. Perancangan Berorientasi Objek

Secara *garis besar*, proses dalam OOD adalah mengubah model yang dihasilkan dari tahap OOA menjadi model desain yang menyediakan *blueprint* untuk konstruksi perangkat lunak. Proses dalam OOD dibagi dalam 5 aktivitas yaitu perancangan sistem, perancangan objek, perancangan antarmuka pengguna, perancangan manajemen data, dan perancangan manajemen fungsionalitas. Bab ini membahas tentang aktivitas-aktivitas yang dilakukan dalam OOD beserta hasil dari aktivitas-aktivitas tersebut.

4.1. Perancangan Sistem

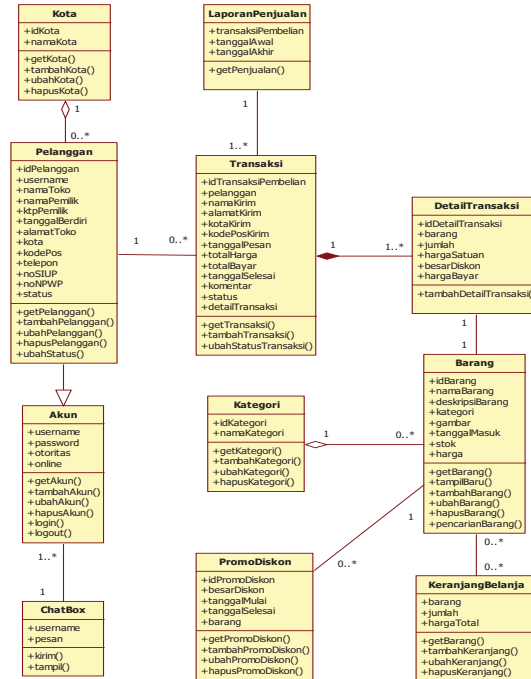
Perancangan sistem mengembangkan detail arsitektural yang dibutuhkan untuk membangun sistem atau produk. Secara garis besar, menjelaskan bagaimana subsistem-subsistem saling berinteraksi, dimulai dengan membagi sistem menjadi subsistem, kemudian menentukan interaksi antarsubsistem. Dalam subbab ini juga ditentukan kelas-kelas yang termasuk dalam setiap subsistem. Interaksi antarsubsistem digambarkan dalam suatu *subsystem collaboration graph* sederhana. Gambar 4.1. menunjukkan kolaborasi antar subsistem yang terdapat dalam AESS.



Gambar 4.1. Subsystem Collaboration Graph AESS

4.2. Perancangan Objek

Perancangan objek membahas tentang perancangan detail objek yang meliputi spesifikasi tipe-tipe atribut, bagaimana operasi-operasi objek tersebut berfungsi, dan interaksi objek yang menghubungkan objek tersebut dengan objek lain. Hasil dari perancangan detail objek dapat dilihat pada *class diagram design*. Gambar 4.2. menjelaskan *class diagram design* dari AESS.



Gambar 4.2. Class diagram design AESS

4.3. Perancangan Antarmuka

Antarmuka pengguna pada AESS berupa form-form yang berisi menu/tombol-tombol (*buttons*). Setiap form memiliki lebih dari satu objek yang terkait dan setiap tombol dapat menjalankan lebih dari satu operasi pada satu objek atau beberapa objek. Sebagai contoh didefinisikan objek pelanggan1 dari kelas Pelanggan dengan antarmuka form registrasi pelanggan. Objek pelanggan1 memiliki atribut dan operasi dari kelas Akun dan Pelanggan seperti yang telah dijelaskan pada tahap analisis berorientasi objek. Aktor yang berinteraksi dengan objek pelanggan1 adalah pengunjung. Ketika pengunjung menekan tombol simpan, objek pelanggan1 akan menjalankan operasi untuk mengeset nilai atribut-atribut objek pelanggan1. Kemudian objek pelanggan1 akan menjalankan operasi untuk menyimpan nilai atribut-atribut tersebut ke dalam basis data. Ilustrasi tersebut menggambarkan konsep antarmuka sebuah objek. Secara umum, tampilan perancangan antarmuka untuk *registrasi pelanggan* menampilkan form untuk mengisi data pelanggan yang dibutuhkan untuk melakukan pendaftaran. Perancangan antarmuka *registrasi pelanggan* ditunjukkan pada gambar 4.3.

Buat Akun

Nama Toko :

Nama Pemilik :

No. SIUP :

No. NPWP :

No. KTP Pemilik :

Tanggal Berdiri :

Alamat Toko :

Kota :

Kode Pos :

Telepon :

Username :

Password :

Konfirmasi Password :

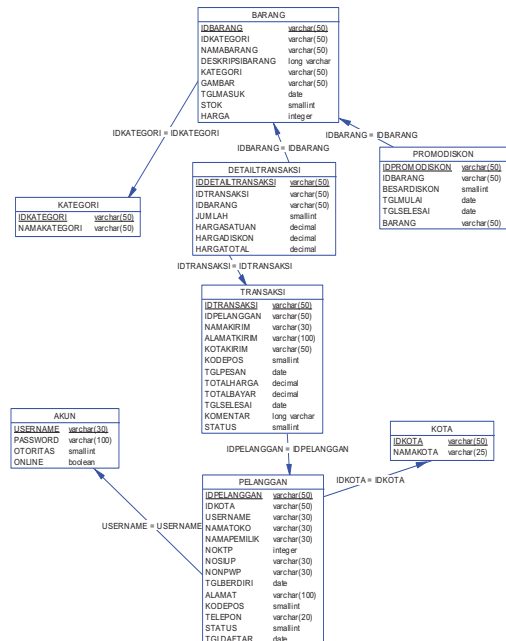
SIMPAN

ULANG

Gambar 4.3. Perancangan antarmuka *registrasi pelanggan*

4.4. Perancangan Manajemen Data

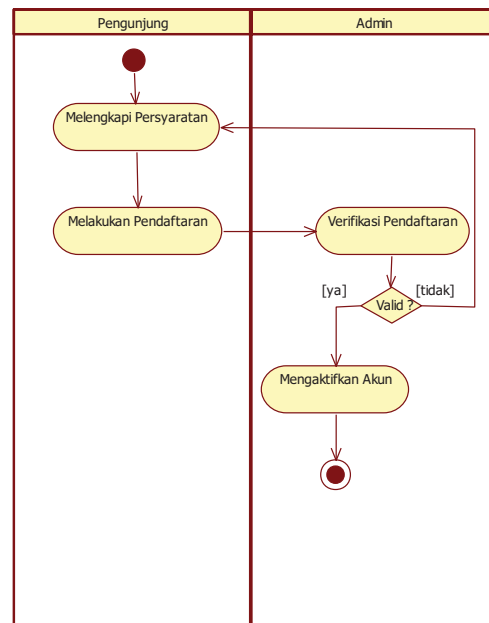
Komponen manajemen data yang dibahas pada subbab ini adalah tentang perancangan basis data dengan menggunakan metode *Mapping*, yaitu mengubah (mapping) model analisis ke dalam basis data relasional dan perancangan operasi-operasi objek atau objek-objek lain yang dapat digunakan untuk mengelola data suatu objek tertentu. Skema Relasional Basis Data AESS ditunjukkan pada gambar 4.4.



Gambar 4.4. Skema relasional basis data AESS

4.5. Perancangan Manajemen Fungsionalitas

Fungsi-fungsi yang terdapat dalam AESS, serta bagaimana aktor berinteraksi dengan fungsi-fungsi tersebut telah ditentukan pada model analisis, yaitu *system use case diagram* dan *use case detail*. Perancangan manajemen fungsionalitas merancang implementasi objek-objek ke dalam fungsi-fungsi yang ada, digambarkan dalam *activity diagram*. *Activity diagram* untuk registrasi pelanggan ditunjukkan pada gambar 4.5.



Gambar 4.5. *Activity diagram* registrasi pelanggan

5. Implementasi dan Pengujian

Tahap implementasi AESS meliputi beberapa hal yaitu implementasi perangkat keras dan perangkat lunak, implementasi kelas, implementasi basis data, dan implementasi antarmuka pengguna.

5.1. Implementasi Perangkat Keras dan Perangkat Lunak

Spesifikasi minimum perangkat keras yang digunakan dalam menjalankan AESS adalah sebagai berikut:

1. CPU : Pentium 4 / 2,80 GHz
2. Memori : 256 MB
3. Harddisk : 40 GB
4. Monitor : 800 x 600 pixel
5. Mouse : mouse wheel 2 button 1 scroll
6. Keyboard : standart keyboard 104 keys.

Sedangkan perangkat lunak yang digunakan dalam menjalankan AEISS sebagai berikut:

1. Sistem operasi : Microsoft Windows XP
2. Aplikasi server : XAMPP 1.7.3
3. Browser : Mozilla Firefox 3, Google Chrome 5, dan Internet Explorer 8.

5.2. Implementasi Kelas

Kelas-kelas yang telah didefinisikan pada tahap analisis dan perancangan berorientasi objek diimplementasikan ke dalam bahasa pemrograman berorientasi objek. Dalam pembuatan AEISS, bahasa pemrograman yang dipakai adalah PHP. Kelas-kelas diimplementasikan dalam suatu *source code* dengan ekstensi *file .php* yang disimpan pada media penyimpanan sebagai *library* kelas yang suatu saat dapat digunakan kembali (*reuse*).

5.3. Implementasi Basis Data

Tabel-tabel yang telah ditentukan pada tahap perancangan berorientasi objek diimplementasikan ke dalam *Database Management System MySQL*. Basis data yang dibuat diberi nama “sharp”. Sedangkan nama-nama tabel telah ditentukan pada tahap perancangan berorientasi objek. Implementasi basis data untuk tabel pelanggan ditunjukkan pada gambar 5.1.

Nama Field	Deskripsi	Type & Length	Null	Keterangan
idPelanggan	Identitas pelanggan sebagai primary key	varchar(50)	NO	Primary Key
username	Username pelanggan	varchar(20)	NO	Foreign Key
namaToko	Nama toko pelanggan	varchar(20)	NO	-
namaPemilik	Nama pemilik toko	varchar(30)	NO	-
noKTP	Nomor KTP pelanggan	int(15)	NO	-
noSIUP	Nomor SIUP pelanggan	varchar(20)	NO	-
noNPWP	Nomor NPWP pelanggan	varchar(20)	NO	-
tglBerdiri	Tanggal berdirinya toko	date	NO	-
alamat	Alamat toko	varchar(50)	NO	-
idKota	id kota toko berada	varchar(50)	NO	Foreign Key
kodepos	Kodepos toko berada	mediumint (5)	NO	-
telepon	Nomor telepon toko	varchar(20)	NO	-
status	Status pelanggan aktif atau tidak aktif	tinyint(1)	NO	-
tgldaftar	Tanggal pendaftaran	date	NO	-

Gambar 5.1. Tabel Pelanggan

5.4. Implementasi Antarmuka

Perancangan antarmuka yang telah dibuat pada tahap perancangan berorientasi objek diimplementasikan ke dalam bahasa pemrograman berorientasi objek dan digunakan sebagai *interface* kelas untuk menjelaskan operasi-operasi kelas. Subbab ini akan menjelaskan implementasi antarmuka pengguna dari tiap *use case detail* dan berdasarkan pada perancangan antarmuka yang telah dibuat pada tahap perancangan. Implementasi antarmuka registrasi pelanggan ditunjukkan pada gambar 5.2.

Buat Akun

* Kami Hanya Melayani Penjualan Kepada Toko yang Memiliki Surat Ijin Usaha Perdagangan

Nama Toko :

Nama Pemilik :

No. SIUP :

No. NPWP :

No. KTP Pemilik :

Tanggal Berdiri :

Alamat Toko :

Kota :

Kode Pos :

Telepon :

Username :

Password :

Konfirmasi Password :

Gambar 5.2. Implementasi antarmuka registrasi pelanggan

5.5. Pengujian

Perangkat lunak yang baik atau sukses dibuat adalah perangkat lunak yang dapat memenuhi spesifikasi yang dibutuhkan oleh pengguna perangkat lunak, memiliki performa yang sempurna dalam jangka waktu yang lama, mudah dimodifikasi dan bahkan mudah untuk digunakan, serta dapat mengubah sesuatu menjadi lebih baik. Pengujian perangkat lunak mengevaluasi dan memastikan apakah perangkat lunak telah memenuhi spesifikasi yang dibutuhkan sehingga dapat dikatakan sukses dibuat.

Berdasarkan hasil pengujian, dapat disimpulkan bahwa AEISS telah memenuhi spesifikasi kebutuhan perangkat lunak yang telah didefinisikan. Dari hasil pengujian, dapat diketahui bahwa AEISS telah memenuhi untuk :

1. Menampilkan barang terbaru, barang berdasarkan kategori, dan melakukan pencarian barang.
2. Melakukan pendaftaran pelanggan.
3. Mengumpulkan barang pada keranjang belanja.
4. Melakukan transaksi pembelian barang.
5. Menyampaikan komentar terhadap transaksi yang dilakukan.
6. Mengubah data pribadi pelanggan.
7. Melakukan verifikasi pelanggan
8. Mengelola data kota.
9. Mengelola data kategori barang.
10. Mengelola data barang.
11. Mengelola data promo.
12. Melakukan konfirmasi pembelian.
13. Mencetak laporan penjualan.
14. Melakukan otentikasi pengguna.

15. Mengubah username/password.
16. Memfasilitasi pelanggan untuk berkomunikasi dengan admin.
17. Melihat data transaksi bagi admin.
18. Melihat riwayat transaksi bagi pelanggan.

6. Penutup

6.1. Kesimpulan

Kesimpulan yang dapat diambil dalam pengerjaan tugas akhir ini adalah dihasilkan sebuah Aplikasi E-commerce yang dapat menampilkan informasi mengenai barang-barang yang dijual pada PT Sharp cabang Semarang secara lengkap dan terkini serta dapat memberikan fungsi transaksi pembelian barang oleh konsumen dengan cepat dan mudah. Selain itu, aplikasi yang dihasilkan dibuat dengan konsep berorientasi objek, sehingga memiliki kualitas lebih baik dilihat dari sisi fungsionalitasnya dan proses pembuatannya

6.2. Saran

Untuk pengembangan lebih lanjut, Aplikasi E-commerce pada PT Sharp cabang Semarang dapat dikembangkan dengan melengkapi cara pembayaran yang lebih banyak pilihannya, mudah, serta aman. Selain itu dapat dikembangkan juga untuk cabang-cabang lain yang tersebar di seluruh Indonesia.

Daftar Pustaka

- [1] _____, *Aplikasi E-commerce di CV Tupai ASH*, Diakses dari http://jiunkpe-ns-s1-2006-26401085-8017-e_commerce-chapter1.pdf pada tanggal 12 November 2010.
- [2] _____, *Profil Perusahaan PT Sharp*, Diakses dari <http://www.sharp-indonesia.com/new/tentang-sharp/profil-perusahaan.html> pada tanggal 12 November 2010.
- [3] _____, *Javascript Guide*, Diakses dari <http://blog.codingwear.com/JavascriptGuide.zip> pada tanggal 12 November 2010.
- [4] Booch G., dkk, 2005, *The Unified Modeling Language User Guide Second Edition*, Westford : Pearson Education, Inc.
- [5] Dewanto I. J., 2006, *Web desain - metode aplikasi dan implementasi*, Yogyakarta : graha ilmu.
- [6] Gutmans, Andi., dkk, 2005, *PHP 5 Power Programming*, Prentice Hall.
- [7] Laudon K. C., 2008, *E-commerce business technology society fourth edition*, Pearson International Edition.
- [8] Lee M., dkk, 2005, *StarUML 5.0 User Guide* Free Software Foundation, <http://www.staruml.com/starumldoc/user-guide/>.
- [9] Liem I., 2003, *Diktat Kuliah Pemrograman Berorientasi Objek*, Bandung : Departemen Teknik Informatika Institut Teknologi Bandung.
- [10] Nugroho, B., 2004, *Aplikasi pemrograman web dinamis dengan PHP dan MySQL*, Yogyakarta : Gava Media.
- [11] Nugroho, B., 2005, *Membuat aplikasi penjualan dengan PHP - MySQL*, Yogyakarta : Andi Offset.
- [12] Pressman R. S., 2001, *Software Engineering : A Practitioner's Approach Fifth Edition*, New York : McGraw - Hill Companies, Inc.
- [13] Prihatna, H., 2005, *Kiat Praktis menjadi webmaster profesional*, Jakarta : Elex media.
- [14] Purbo O. W. dan Wahyudi A. A., 2001, *Mengenal E-commerce*, Jakarta : PT Elex Media Komputindo.
- [15] Tampubolon, S., 2003, *Aspek Hukum Nama Domain Di Internet*, Jakarta : PT Tatanusa.
- [16] Sommerville I., 2001, *Software Engineering 6th Edition*, Addison Wesley.
- [17] Suehring S., 2002, *MySQLTM Bible*, New York : Wiley Publishing, Inc.
- [18] Sulhan M., 2007, *Pengembangan aplikasi berbasis web dengan PHP & ASP*, Yogyakarta : gava media.
- [19] Sutarman., 2003, *Membangun aplikasi web dengan PHP dan MySQL*, Yogyakarta : Graha ilmu.
- [20] Wardani, Ruly A. 2010. *Aplikasi Pengolahan dan Publikasi Nilai Siswa Studi Kasus SMKN 2 Salatiga*. Semarang : Universitas Diponegoro

